

# Application support with libcamera



Embedded Linux Conference  
Dublin 2022

Kieran Bingham  
[kieran.bingham@ideasonboard.com](mailto:kieran.bingham@ideasonboard.com)



+ - / \ - +  
| (o) |  
+ - - - - +

- Complex Cameras
- Complications
- Existing Solutions
- Community Support
- Future Developments
- Q+A



- This talk references many external projects, packages, and software.
- All logos, trademarks, and other identifying marks belong to their respective owners.
- In many places I'm talking about work that /other/ people have done or contributed
  - No intention or desire to take credit for their work



**Legalese...**



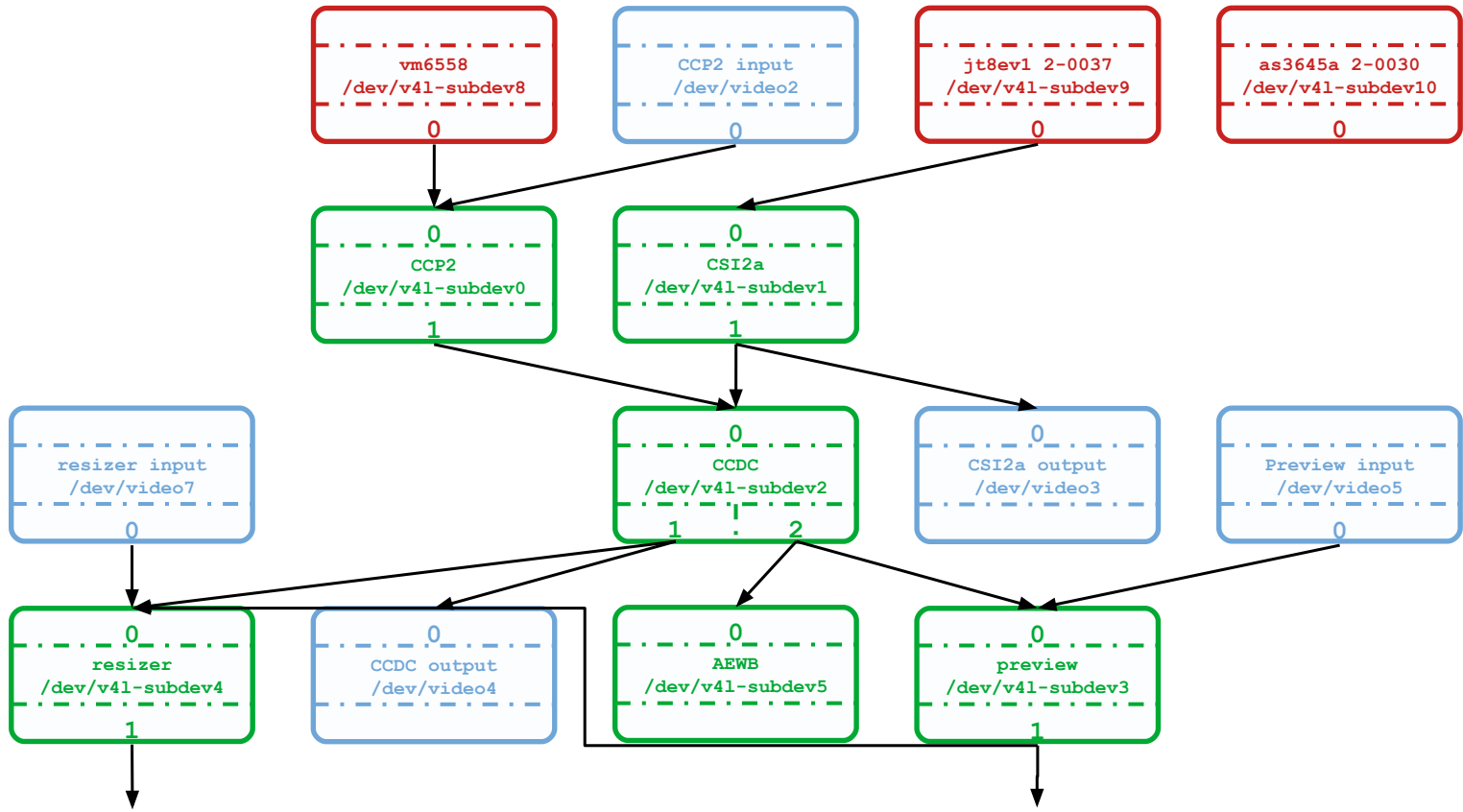
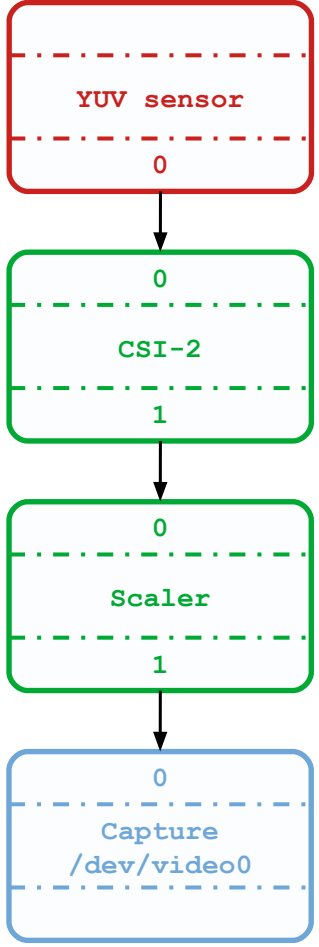
- **Complex Cameras**
- **Complications**
- **Existing Solutions**
- **Community Support**
- **Future Developments**
- **Q+A**



C s  
a e  
m n  
e s  
r o  
a r

S t  
o n  
C t  
e  
c r  
a f  
m a  
e c  
a e

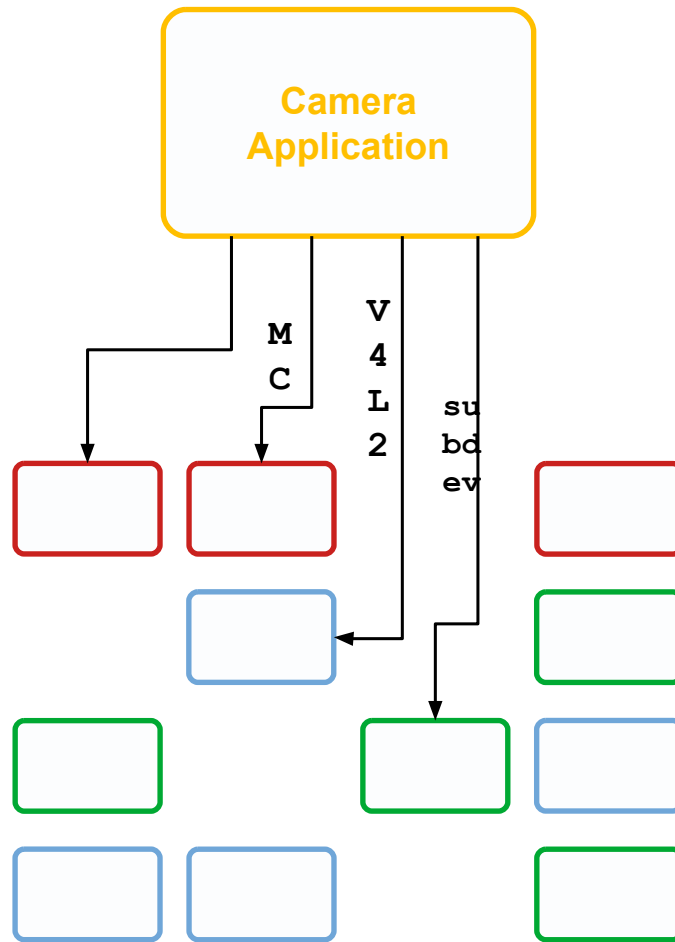
A P  
I

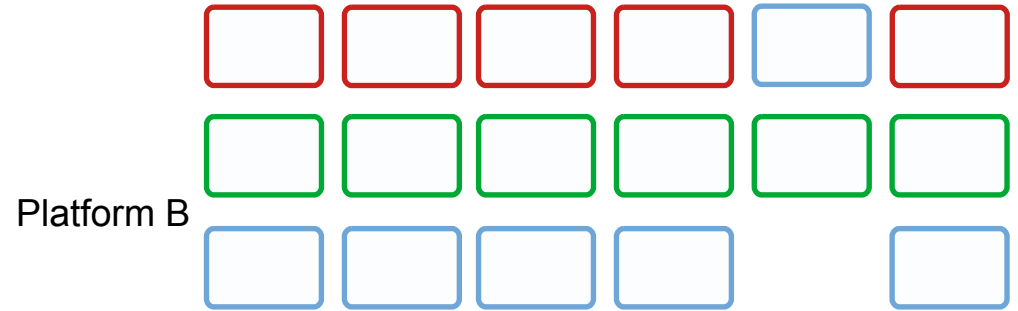
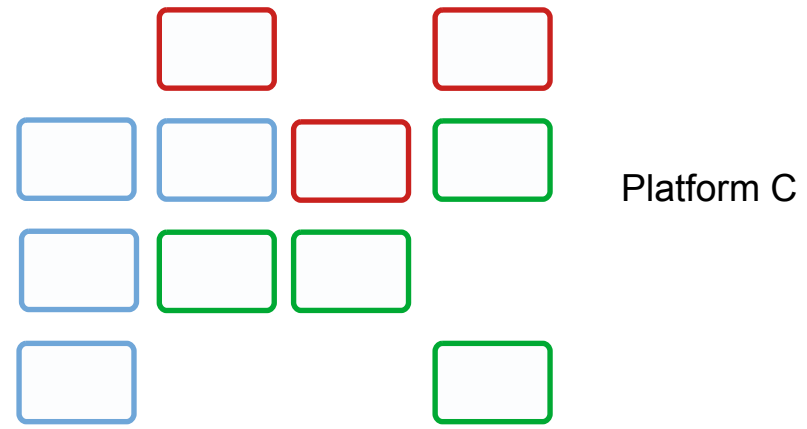
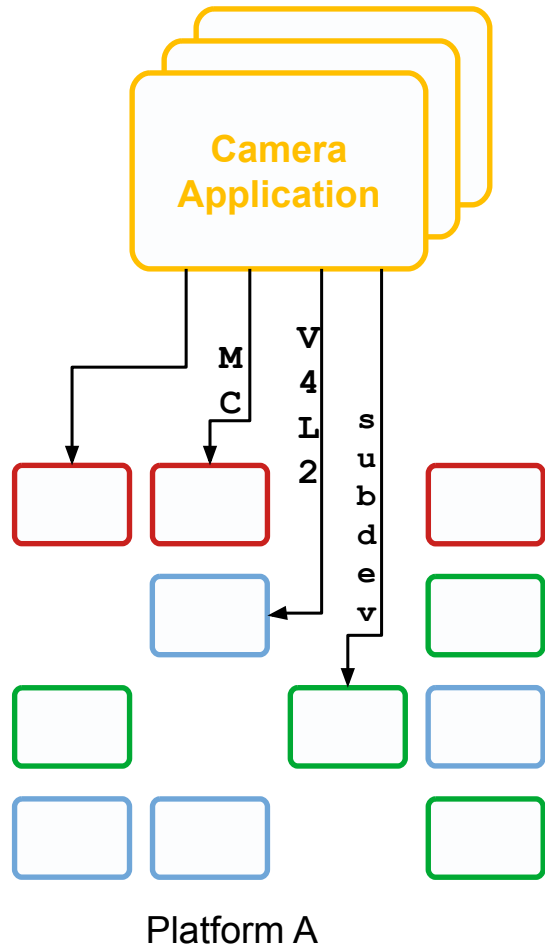


OMAP3 Camera in Nokia N900 - 2009

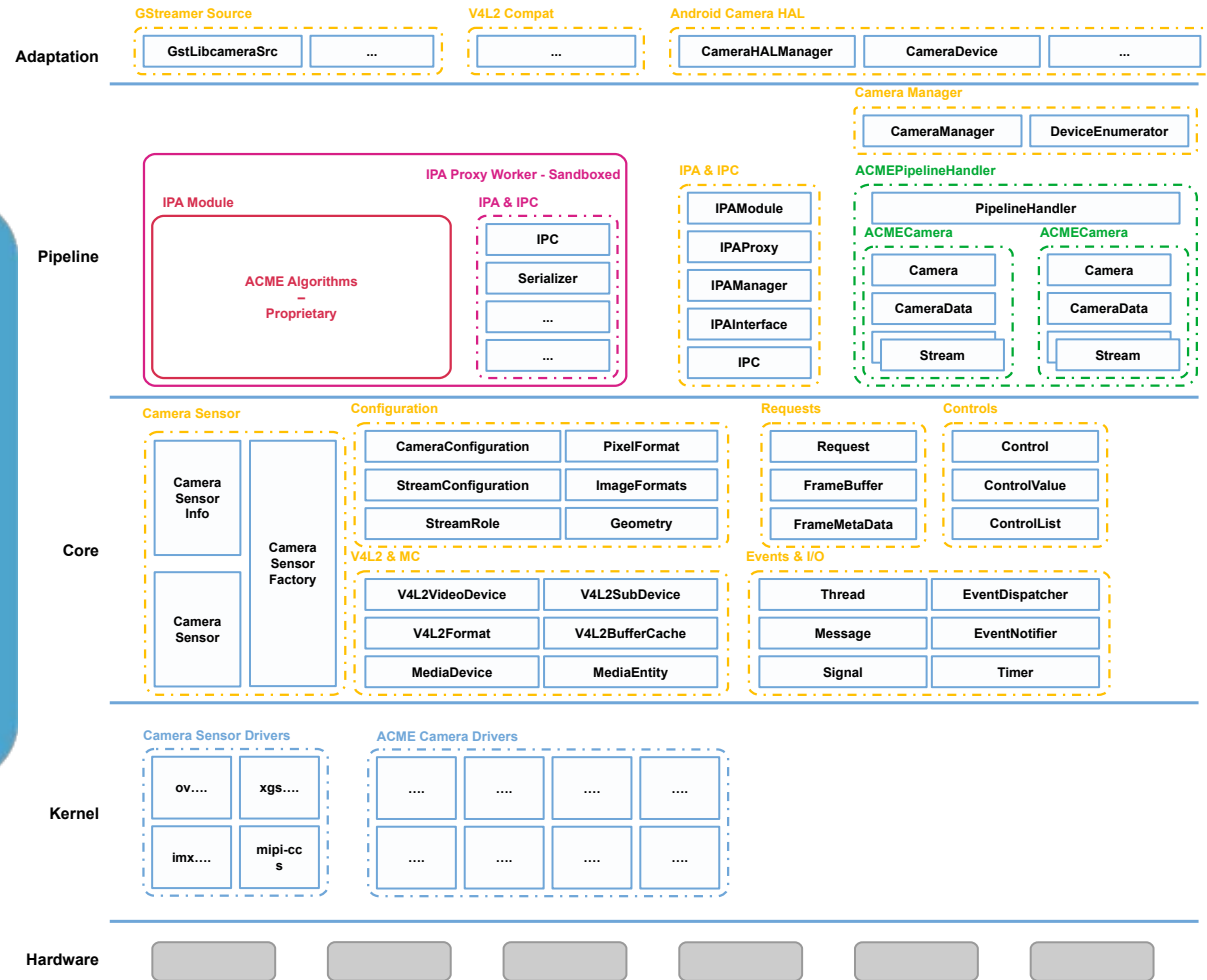


Cameras are Complex





**But it doesn't scale**



**libcamera fills that gap**

**IDEAS  
ON BOARD**





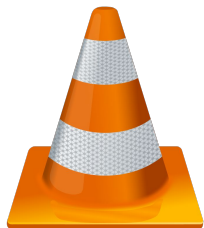
- Complex Cameras
- **Complications**
- Existing Solutions
- Community Support
- Future Developments
- Q+A



- V4L2 started between 1998, 2002
- Consistent API supports many Video capture devices
  - Wide support of pixel formats ...
  - Same API for (existing, simple) Cameras, Digital TV DVB, Set Top Box ...
- It exists
  - There hasn't been anything else at the kernel level (upstream)
- Widely used (thoroughly tested)



**Everybody loves V4L2 \***



Media/Camera Applications



Multimedia and Application Frameworks



Browsers



WebRTC



Conferencing Utilities



**V4L2 is already used everywhere**

- Really - they love '/dev/video0'
- Media Controller
  - Entities, links, pads, format negotiation (propagation)
- Subdevices
  - Direct control over specific internal components
  - Which one do you configure?
- Multiple video nodes for a single "Camera device"
  - UVC - 'Metadata video node'
  - CSI2 Receiver
  - ISP - Statistics, Parameter Buffers, Multiple image streams
  - M2M Dewarper



**Not everybody loves V4L2 \***

- 3A algorithms need to be handled in userspace
  - Crucial for RAW sensors. YUV sensors are becoming obsolete
- Laptops are now using complex cameras
  - Dell, Lenovo, HP, Surface ...
- Embedded devices already use complex cameras
  - OEM/ODM ... need custom solutions to manage each camera
- No portable mobile camera applications.
  - Mobile is dominated by Android, with mostly binary camera stacks



**V4L2 alone isn't enough**



<https://www.emersonhc.com/change-management/people-hard-wired-resist-change>

- Many applications don't yet support libcamera
- Adding the support takes effort
  - maintainers haven't expected this
- C applications don't want to use C++
  - People can be scared of the ++
- Is it even finished yet?
  - Releases, ABI stability ...



**But ... now there's a new API to use**



- Complex Cameras
- Complications
- **Existing Solutions**
- Community Support
- Future Developments
- Q+A



Gstlibcamerasrc brings the whole Gstreamer ecosystem to libcamera devices:

- Encoding / Streaming
- Composing / Mixing
- Audio

Camera Viewer

```
gst-launch-1.0 libcamerasrc ! 'video/x-raw,width=1280,height=720' ! glimagesink
```

JPEG Network streamer

```
gst-launch-1.0 libcamerasrc ! \
    video/x-raw,colorimetry=bt709,format=NV12,width=1280,height=720,framerate=30/1 ! \
    jpegenc ! multipartmux ! \
    tcpserver sink host=0.0.0.0 port=5000
```

JPEG Network Receiver

```
gst-launch-1.0 tcpclientsrc host=$DEVICE_IP port=5000 ! \
    multipartdemux ! jpegdec ! autovideosink
```



**libcamera provides a gstreamer element**



Raspberry Pi is transitioning from a legacy camera software stack based on proprietary Broadcom GPU code to an open-source stack based on `libcamera`. Raspberry Pi OS images from *Bullseye* onwards will contain **only** the `libcamera`-based stack. Raspberry Pi OS images up to and including *Buster* will contain the legacy *Raspicam* stack, though the `libcamera` stack and applications can be installed using *apt*, or built by following the *normal build instructions*.

Users are encouraged to use the newest OS images and the `libcamera`-based stack because:

- It will continue to be developed moving forward.
- Raspberry Pi and 3rd parties can fix bugs and problems in the camera stack.
- Raspberry Pi and 3rd parties can add new features to the camera stack.
- It is much easier to add support for new cameras.
- 3rd parties can add support directly for their own cameras.
- Nearly all aspects of the camera tuning can be changed by users.
- It integrates much more conveniently with other standard Linux APIs.
- Raspberry Pi supply a set of `libcamera-apps` which emulate most of the features of the legacy applications.
- It provides a feature-rich post-processing framework integrating OpenCV and TensorFlow Lite.
- Libcamera makes it easier to control the parameters of the image sensor and the camera system.
- It is fully supported on 64-bit operating systems.

<https://www.raspberrypi.com/documentation/accessories/camera.html>



# Raspberry Pi provide their own implementation

- Few users like C++
- Python is “friendly”
- Fast to prototype
- ‘Picamera2’ simplifies the libcamera python API

```
#!/usr/bin/python3
```

```
# Capture a JPEG while still running in the preview mode. When you  
# capture to a file, the return value is the metadata for that image.
```

```
import time
```

```
from picamera2 import Picamera2, Preview
```

```
picam2 = Picamera2()
```

```
preview_config = picam2.create_preview_configuration(main={"size": (800, 600)})  
picam2.configure(preview_config)
```

```
picam2.start_preview(Preview.QTGL)
```

```
picam2.start()
```

```
time.sleep(2)
```

```
metadata = picam2.capture_file("test.jpg")
```

```
print(metadata)
```

```
picam2.close()
```



## We have python support

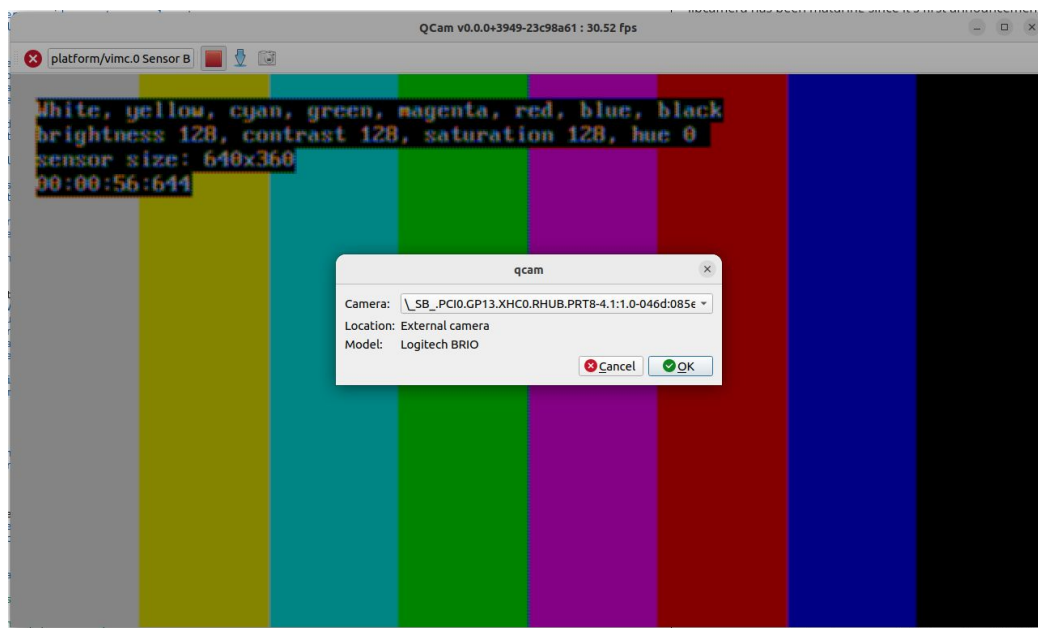


# An android HAL implementation

- LD\_PRELOAD solution
  - LIBCAMERA\_PUBLIC int **open**(const char \*path, int oflag, ...)
  - LIBCAMERA\_PUBLIC int open64(const char \*path, int oflag, ...)
  - LIBCAMERA\_PUBLIC int openat(int dirfd, const char \*path, int oflag, ...)
  - LIBCAMERA\_PUBLIC int \_\_openat\_2(int dirfd, const char \*path, int oflag)
  - LIBCAMERA\_PUBLIC int openat64(int dirfd, const char \*path, int oflag, ...)
  - LIBCAMERA\_PUBLIC int **dup**(int oldfd)
  - LIBCAMERA\_PUBLIC int **close**(int fd)
  - LIBCAMERA\_PUBLIC void \***mmap**(void \*addr, size\_t length, int prot, int flags, int fd, off\_t offset)
  - LIBCAMERA\_PUBLIC void \*mmap64(void \*addr, size\_t length, int prot, int flags, int fd, off64\_t offset)
  - LIBCAMERA\_PUBLIC int **munmap**(void \*addr, size\_t length)
  - LIBCAMERA\_PUBLIC int **ioctl**(int fd, unsigned long request, ...)
- ‘libcamerify’
  - \$ libcamerify -d -d myV4L2Application -myArgs

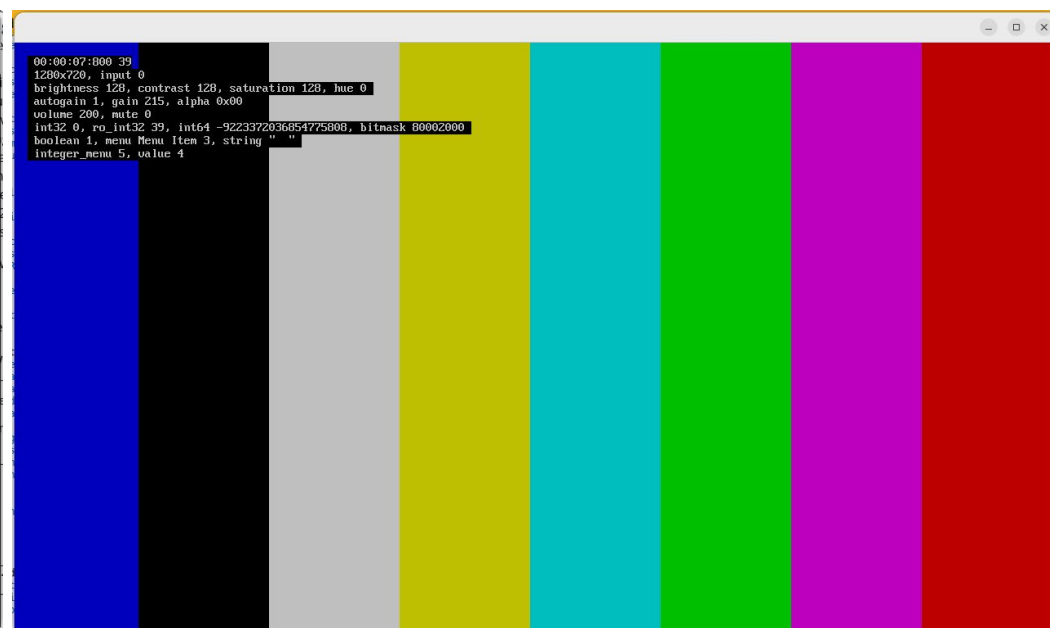


... and a V4L2 compatibility layer



\$ qcam -r gles

(Using VIMC)



\$ cam -c3 -C -S --stream pixelformat=YUYV

(Using VIVID)



# As well as test applications



```

159 int main()
160 {
161     /*
162     * -----
163     * Create a Camera Manager.
164     *
165     * The Camera Manager is responsible for enumerating all the Camera
166     * in the system, by associating Pipeline Handlers with media entities
167     * registered in the system.
168     *
169     * The CameraManager provides a list of available Cameras that
170     * applications can operate on.
171     *
172     * When the CameraManager is no longer to be used, it should be deleted.
173     * We use a unique_ptr here to manage the lifetime automatically during
174     * the scope of this function.
175     *
176     * There can only be a single CameraManager constructed within any
177     * process space.
178     */
179     std::unique_ptr<CameraManager> cm = std::make_unique<CameraManager>();
180     cm->start();
181
182     /*
183     * Just as a test, generate names of the Cameras registered in the
184     * system, and list them.
185     */
186     for (auto const &camera : cm->cameras())
187         std::cout << " - " << cameraName(camera.get()) << std::endl;

```

```

189 /*
190 * -----
191 * Camera
192 *
193 * Camera are entities created by pipeline handlers, inspecting the
194 * entities registered in the system and reported to applications
195 * by the CameraManager.
196 *
197 * In general terms, a Camera corresponds to a single image source
198 * available in the system, such as an image sensor.
199 *
200 * Application lock usage of Camera by 'acquiring' them.
201 * Once done with it, application shall similarly 'release' the Camera.
202 *
203 * As an example, use the first available camera in the system after
204 * making sure that at least one camera is available.
205 *
206 * Cameras can be obtained by their ID or their index, to demonstrate
207 * this, the following code gets the ID of the first camera; then gets
208 * the camera associated with that ID (which is of course the same as
209 * cm->cameras()[0]).
210 */
211 if (cm->cameras().empty()) {
212     std::cout << "No cameras were identified on the system."
213         << std::endl;
214     cm->stop();
215     return EXIT_FAILURE;
216 }
217
218 std::string cameraId = cm->cameras()[0]->id();
219 camera = cm->get(cameraId);
220 camera->acquire();

```

<https://git.libcamera.org/libcamera/simple-cam.git/>

**IDEAS**  
ON BOARD

# And a sample 'hello world' for the API

```
kbingham@Monstersaurus: ~/iob/libcamera/ci/libcamera-ci
Found ninja-1.10.1 at /usr/bin/ninja
ninja: Entering directory `/home/kbingham/iob/libcamera/ci/libcamera-ci/builds/60-simple-cam.sh'
[1/3] Compiling C++ object simple-cam.p/event_loop.cpp.o
[2/3] Compiling C++ object simple-cam.p/simple-cam.cpp.o
[3/3] Linking target simple-cam
[143:25:43.354168806] [2444412] INFO IPAManager ipa_manager.cpp:141 libcamera is not installed. Adding '/home/kbingham/iob/libcamera/ci/libcamera-ci/builds/unit-tests/src/ipa' to the IPA search path
[143:25:43.356069467] [2444412] INFO Camera camera_manager.cpp:293 libcamera v0.0.0+3902-6225d647
[143:25:43.369636898] [2444415] WARN CameraSensorProperties camera_sensor_properties.cpp:174 No static properties available for 'Sensor B'
[143:25:43.369672565] [2444415] WARN CameraSensorProperties camera_sensor_properties.cpp:176 Please consider updating the camera sensor properties database
[143:25:43.369694136] [2444415] WARN CameraSensor camera_sensor.cpp:411 'Sensor B': Failed to retrieve the camera location
[143:25:43.371991431] [2444415] INFO IPAProxy ipa_proxy.cpp:130 libcamera is not installed. Loading IPA configuration from '/home/kbingham/iob/libcamera/ci/libcamera-ci/src/libcamera/src/ipa/vimc/data'
- 'Logitech BRIO' (\_SB_.PCI0.GP13.XHC0.RHUB.PRT8-4.1:1.0-046d:085e)
- (platform/vimc.0 Sensor B)
Default viewfinder configuration is: 1920x1080-NV12
Validated viewfinder configuration is: 1920x1080-NV12
[143:25:43.374254333] [2444412] INFO Camera camera.cpp:1026 configuring streams: (0) 1920x1080-NV12
Allocated 4 buffers for stream

Request completed: Request(0:C:0/1:0)
    SensorTimestamp = 516344180492000
seq: 000000 timestamp: 516344180492000 bytesused: 2073600/1036800

Request completed: Request(1:C:0/1:0)
    SensorTimestamp = 516344404491000
seq: 000001 timestamp: 516344404491000 bytesused: 2073600/1036800

Request completed: Request(2:C:0/1:0)
    SensorTimestamp = 516344432527000
```



## And a sample 'hello world' for the API



**libcamera** @libcamera · May 26

A surface go 2, running chromium browser through the gnome camera portal, through [@PipewireP](#) , and through [@libcamera](#) !!!

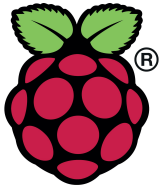


# Pipewire integration brings desktop use cases

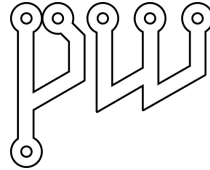
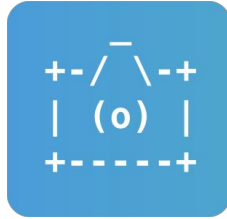




XDG-Camera-Portal



Raspberry Pi



Jitsi Meet



<https://flatpak.github.io/xdg-desktop-portal/#gdbus-org.freedesktop.portal.Camera>



Such as video conferencing through chromium



# Open Source Cameras With libcamera Technical Showcase

Kieran Bingham / Jacopo Mondi / Dan Scally  
Ideas on Board

CE Workgroup Linux Foundation / Embedded Linux Conference



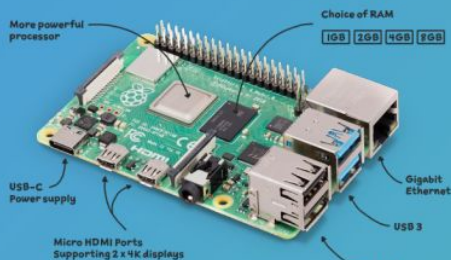
## Implementation Showcase: An Open Source Camera Stack

- Modern userspace API that fully abstracts kernel drivers
- Desktop, Embedded, Android & Chrome OS
- Automatic device discovery and configuration
- Imaging algorithms (3A, lens shading, denoising, dewarping, ...)



## Integration Showcase: An Open Source Video Call

- Fully open-source on both platforms: camera drivers (Linux), imaging algorithms (libcamera), media server (PipeWire), and web browser (Chromium)
- libcamera and PipeWire latest top-of-tree
- Chromium with PipeWire integration (in development by Pengutronix)



Raspberry Pi 4



Intel IPU3

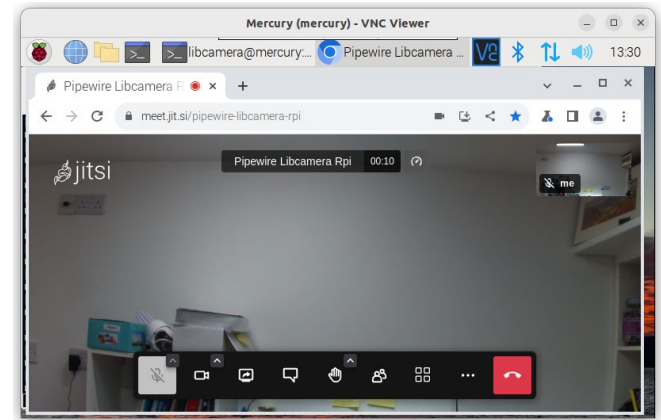


More information: <https://libcamera.org/entries/2022-09-14.html>

IDEAS  
ON BOARD



- Lacks correct format negotiation
- Stride not correctly managed
  - Affects frame sizes that are not a multiple of 32, in width. 1280x720 is usable
- ~~NV12 support has format configuration faults in WebRTC~~
- PipeWire issue with multiple cameras.
  - Fixed to single camera for the moment
- Segmentation faults with RPi chromium build with the V4L2 M2M decoders.
  - A pain - but not related to the camera work



**But this is still in development and has a way to go**



- Complex Cameras
- Complications
- Existing Solutions
- **Community Support**
- Future Developments
- Q+A



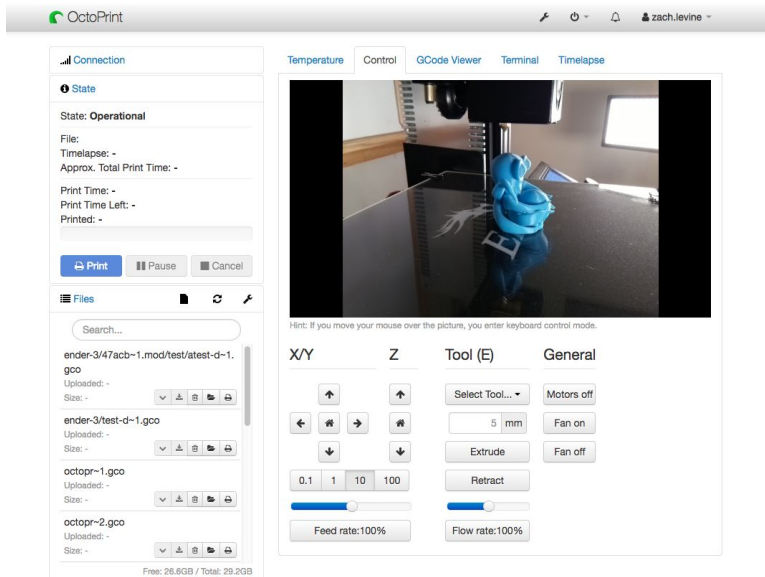




- <https://github.com/Motion-Project/motion/>
- Widely used with Raspberry Pi devices
- Working with 'libcamerify'
- Proposed native libcamera integration - but C++ rejected
- <https://github.com/Motion-Project/motionplus>
  - "MotionPlus is a break at version 4.2.2 from the Motion application. MotionPlus removes some of the outdated processes and features of the Motion application and introduces new functionalities.
  - Could be suitable for someone to write native libcamera integration



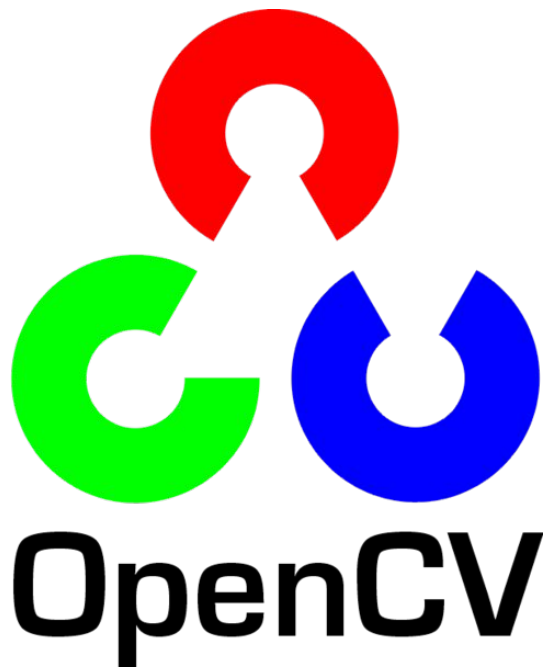
**Motion**



- <https://sourceforge.net/projects/mjpg-streamer/>
- <https://github.com/jacksonliam/mjpg-streamer>
- MJPG-streamer takes JPGs from Linux-UVC compatible webcams, filesystem or other input plugins and streams them as M-JPEG via HTTP to webrowsers, VLC and other software. It is the successor of uvc-streamer, a Linux-UVC streaming application with Pan/Tilt
- <https://github.com/ArduCAM/mjpg-streamer>



# mjpeg-streamer



- libcamerify
  - Frame rate support to be added to the V4L2 adaptation layer.
  - [libcamera-devel,v3] v4l2: Support setting frame rate in the V4L2 Adaptation layer
  - <https://patchwork.libcamera.org/patch/15392/>
- Likely a good candidate for a direct libcamera implementation
  - GSoC open
- GStreamer pipeline support already possible through the gstlibcamerasrc

- <https://github.com/ericcurtin/twincam>
  - A lightweight camera application, designed to start quickly in a bare environment. It is named twincam as it is built with automotive in mind like a twin-cam engine, it is simply the name of the application.
- <https://github.com/folkertvanheusden/constatus>
  - Constatus monitors, converts, transforms, filters and multiplexes video-feeds. Feeds like IP-cameras, "video4linux"-devices, pixelflut, VNC-servers, Raspberry Pi-cameras, etc. It is an NVR (network video recorder) with special features.

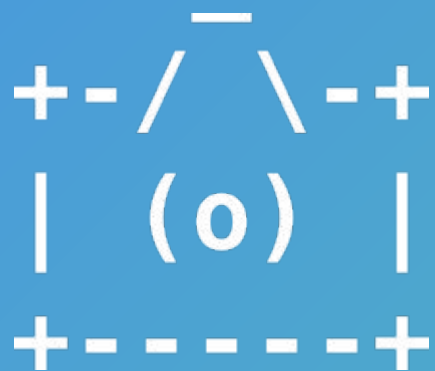
#### Potential (upcoming?) Users

- Megapixels / Millipixels
  - Pinephone ... Custom 'ini' files to handle media controller.
  - Can't manage an ISP (Pinephone Pro, RK3399, ... )
- Gnome Camera App
  - 'Libaperture'
- Your Camera based app here ...
  - Please come and talk to me if you have a use case/app already.
  - I want to hear about more users and use cases!
- #libcamera on irc.oftc.net / Matrix Bridge / Mailing list.



## And more users, or potential users ...





- Complex Cameras
- Complications
- Existing Solutions
- Community Support
- **Future Developments**
- Q+A



The pain of loss is greater than  
the power of gain.

To overcome the resistance of change:

- **Dissatisfaction** with the way things are now
- A positive **vision** of the future
- Concrete **steps** to make the vision a reality



<https://www.emersonhc.com/change-management/people-hard-wired-resist-change>

<https://www.emersonhc.com/change-management/people-hard-wired-resist-change>

**Applications (developers) need to see the benefits**



- libwebrtc changes need more review and testing
- Many common use cases will be opened up
- Pengutronix are already leading the way on this

#### **WebRTC**

Issue 13177: Use Pipewire for camera/webcam access as well : <https://bugs.chromium.org/p/webrtc/issues/detail?id=13177>

Gerrit: Split out generic portal / pipewire code : <https://webrtc-review.googlesource.com/c/src/+263721>

Gerrit: Add pipewire/portal video capture support : <https://webrtc-review.googlesource.com/c/src/+261620>

Gerrit: Add callback for raw frames for video capture : <https://webrtc-review.googlesource.com/c/src/+264548>

#### **Chromium**

Video Capture Linux: add backend for portal / pipewire cameras : <https://chromium-review.googlesource.com/c/chromium/src/+3308882>

Video Capture Linux: factor out v4l2 camera support : <https://chromium-review.googlesource.com/c/chromium/src/+3634526>



# Chromium and libwebrtc needs to be upstream

- Which version of libcamera should we use?
  - ./utils/gen-version.sh
  - 0.0.0+3943-8c458ba3-dirty (2022-09-13T15:08:03+00:00)
- We're still actively developing. We can't yet offer ABI/API stability. ?
- Distributions are /screaming/ for a 'tag'
- We need more (available) CI
  - Freedesktop ?



**libcamera needs to make a release**

- Collabora
- Redhat
- Pengutronix
- Raspberry Pi
- NXP
- ChromeOS



Thank you



- Camera's Complexities have led to libcamera development becoming a necessity on consumer devices (laptops, linux-phones)
- Application support with libcamera is increasing but still a long way to go.
- Pipewire support can already provide desktop integration
  - device security through the XDG Camera Portal
- Python bindings are now available.
  - picamera2 now released by RPi
- GStreamer support has had a lot of improvements



**TL;DR**

? !



libcamera



libcamera







**<https://www.libcamera.org>**

**[kieran.bingham@ideasonboard.com](mailto:kieran.bingham@ideasonboard.com)**



**Contact**



# Thank you!



libcamera



libcamera

